

Université **IBM i**

7 novembre 2023

IBM Innovation Studio Paris

S19 – Apprivoisez l'IFS avec les services SQL

16:00 - 17:00

Philippe Bourgeois

IBM France

pbourgeois@fr.ibm.com

 **infrasdufutur**

#ibmi

#uui2023

#infrastructuredufuturIBM23



Infrastructures du futur



7 et 8 novembre 2023

Objectif de cette présentation

- Faire un point sur les possibilités offertes par les services **SQL IFS**, afin de pouvoir répondre à ce type de questions :
 - Quels sont les 5 plus gros fichiers sous le répertoire /home ?
 - Quels sont les fichiers JSON du répertoire /pb auxquels on a accédé après le 1er octobre 2023 ?
 - Quels sont les fichiers dans /tmp qui n'ont pas été utilisés depuis au moins 2 ans et comment les supprimer ?
 - Quel est le nombre de fichiers et la taille globale de chaque répertoire de l'IFS ?
 - Quels sont les objets de mon répertoire sous /home dont je ne suis pas propriétaire et comment corriger le problème ?
 - Quels sont les objets de /home/bourgeois qui ne sont pas en *PUBLIC *EXCLUDE et comment corriger le problème ?
 - Quels sont les travaux qui ont un verrouillage ou une référence sur le répertoire /home/bourgeois ?
 - Quels sont les fichiers JSON dans /pb qui contiennent "1984"
 - Est-ce que les fichiers listefilms.json qui se trouvent dans 2 répertoires différents ont le même contenu ?
 - Comment lire et décomposer un fichier JSON, XML ou CSV ?
 - Comment copier un fichier IFS d'un répertoire à un autre ?
 - Comment aller chercher un document sur le Web et l'ajouter à un répertoire IFS ?

Agenda

- 1. Services SQL – Introduction

- 2. Services SQL et IFS
 - Contenu de répertoires IFS
 - Droits sur les répertoires et les fichiers IFS
 - Références et verrouillages de fichiers IFS
 - Renommage de fichiers IFS
 - Partages NetServer
 - Lecture et écriture de fichiers IFS (CSV, XML, JSON...)

- 3. Pour aller plus loin

The IBM i logo is displayed in a bold, blue, sans-serif font. It is positioned to the left of the main title and is surrounded by several 3D blue cubes of varying sizes and orientations, some appearing to float or be stacked.

IBM i

1. Services SQL – Introduction

Les services SQL

- Accès en **SQL** aux **fonctions système** de l'IBM i et de DB2 for i
 - Alternative moderne et standard aux commandes CL et aux APIs

- Accessibles sous forme de vues, procédures et fonctions SQL dans les bibliothèques QSYS2, SYSTOOLS et SYSIBMADM

- Deux types :
 - Services IBM i : <https://www.ibm.com/docs/fr/i/7.5?topic=optimization-i-services>
 - Services DB2 for i : <https://www.ibm.com/docs/fr/i/7.5?topic=optimization-db2-i-services>

- Nouveaux services et évolution des services existants à chaque TR (Technology Refresh)

- Disponibles de base dans l'OS ou par mise à jour du groupe de PTFs DB2

Services IBM i

- Sont classés en 18 catégories - <https://www.ibm.com/docs/fr/i/7.5?topic=optimization-i-services>

IBM i Services

Last Updated: 2023-10-10

There are many system services that can be accessed through system-provided SQL views, procedures, and functions. These provide an SQL interface to access, transform, order, and subset the information without needing to code to a system API.

Information about the supported IBM® i releases, including the database PTF groups where each service was introduced or enhanced, is available in the IBM i Technology Updates wiki. See this page for all the details: <https://www.ibm.com/support/pages/node/1119123>

– **Application Services**

These procedures, functions, and views provide interface information that can be used by applications.

– |< **Backup and Recovery Services** >|

These table functions and views provide information about backup and recovery.

– **Communication Services**

These services provide communication information.

– |< **Configuration Services** >|

These services provide configuration information.

– **IFS Services**

These services provide information about the integrated file system.

– **Java Services**

This view and procedure provide Java™ information and JVM management options.

– **Journal Services**

These services provide information about audit journals and data journals.

– **Librarian Services**

These services provide object and library list information.

– **Message Handling Services**

These views and functions provide system message information.

– **Performance Services**

These services provide information related to system performance.

– **PowerHA Services**

These table functions and views provide information about PowerHA®.

– **Product Services**

These services provide information about licensed products.

– **PTF Services**

These services provide PTF and problem reporting information.

– **Security Services**

These views, procedures, and functions provide security information.

– **Spool Services**

These views and functions provide information about spooled files.

– **Storage Services**

These views provide information about storage and storage devices.

– **System Health Services**

For the most important system resources, the IBM i operating system automatically tracks the highest consumption and consumers.

– **Work Management Services**

These views and functions provide system value and job information.

Services IBM i – Comment les obtenir ?

- Disponibles de base dans l'IBM i ou par mise à jour du groupe de PTFs DB2
 - <https://www.ibm.com/support/pages/node/1119123>

IBM i Service	Type of Service	IBM i 7.5	IBM i 7.4	IBM i 7.3	IBM i 7.2
IFS Services					
QSYS2.IFS_JOB_INFO()	Table function	Base	SF99704 Level 4	SF99703 Level 16	Not Supported
QSYS2.IFS_OBJECT_LOCK_INFO()	Table function	Base	SF99704 Level 4	SF99703 Level 16	Not Supported
QSYS2.IFS_OBJECT_PRIVILEGES()	Table function	Base	SF99704 Level 7	SF99703 Level 18	Not Supported
QSYS2.IFS_OBJECT_REFERENCES_INFO()	Table function	Base	SF99704 Level 4	SF99703 Level 16	Not Supported
QSYS2.IFS_OBJECT_STATISTICS()	Table function	Base Enhanced: SF99950 Level 4	SF99704 Level 4 Enhanced: SF99704 Level 13	SF99703 Level 16 Enhanced: SF99703 Level 24	Not Supported
QSYS2.IFS_READ()	Table function	Base	SF99704 Level 10	SF99703 Level 22	Not Supported
QSYS2.IFS_READ_BINARY()	Table function	Base	SF99704 Level 10	SF99703 Level 22	Not Supported
QSYS2.IFS_READ_UTF8()	Table function	Base	SF99704 Level 10	SF99703 Level 22	Not Supported
QSYS2.IFS_WRITE()	Procedure	Base	SF99704 Level 10	SF99703 Level 22	Not Supported
QSYS2.IFS_WRITE_BINARY()	Procedure	Base	SF99704 Level 10	SF99703 Level 22	Not Supported
QSYS2.IFS_WRITE_UTF8()	Procedure	Base	SF99704 Level 10	SF99703 Level 22	Not Supported
QSYS2.SERVER_SHARE_INFO	View	Base Enhanced	SF99704 Level 10	SF99703 Level 22	Not Supported
SYSTOOLS.IFS_RENAME	Scalar function	SF99950 Level 5	SF99704 Level 26	Not Supported	Not Supported

Rappels SQL : la clause WITH

- La clause **WITH** permet de créer des **CTE** (Common Table Expression) :
 - Caractéristiques
 - Instruction **SELECT** pour définir dynamiquement un jeu de données temporaire en mémoire
 - Est disponible le temps de la requête
 - Plusieurs CTE possibles par requête
 - Peuvent se référencer entre elles, dans l'ordre de leur déclaration
 - Utilisation :
 - Diviser une requête complexe en plusieurs étapes
 - Amélioration de la lisibilité
 - Réduction de l'usage de tables de travail
 - Réutilisation d'ensembles de données au sein d'une requête
 - SQL récursif

- Syntaxe :

```

WITH temp1 (z1, z2, z3) AS (SELECT c1, c2, c3 FROM t1 WHERE...),
temp2 AS (SELECT z1, COUNT(*) AS nb FROM
temp1 GROUP BY z1)
SELECT * FROM temp2 ;
  
```

Rappels SQL : exécution de commandes CL en SQL

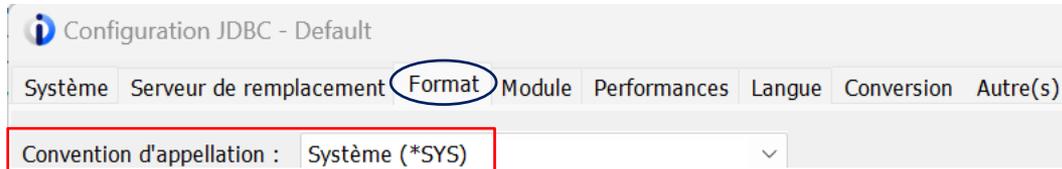
- 1. **Procédure** stockée **QCMDEXC**
 - `CALL QCMDEXC('commande_CL ')`
 - Appelable uniquement par CALL – Ne peut pas être utilisée dans un SELECT

- 2. **Fonction** scalaire (UDF) **QCMDEXC**
 - Peut être utilisée dans un SELECT
 - `SELECT z1, QCMDEXC('commande_CL ') , Z2... FROM ...`
 - La commande CL est exécutée pour chacune des lignes du SELECT
 - Renvoie :
 - **1** si la commande CL s'est bien exécutée
 - **-1** si une erreur s'est produite

Dans les exemples qui suivent

- Dans ACS :
 - Dans le but d'améliorer la lisibilité des requêtes dans le support de cours (en évitant de qualifier systématiquement les fonctions par QSYS2 ou SYSTOOLS) :
 - SELECT ... FROM TABLE(`ifs_object_statistics(...)`)
à la place de SELECT ... FROM TABLE(`qsys2.ifs_object_statistics(...)`)

- 1. J'ai utilisé la convention de nommage *SYS pour utiliser la LIBL (qui contient QSYS2) :



- 2. J'ai ajouté SYSTOOLS à la LIBL : **CL:ADDLIBLE** LIB(SYSTOOLS);



IBM i



Services SQL et IFS

Services IFS

- <https://www.ibm.com/docs/fr/i/7.5?topic=services-ifs>

IFS Services

Last Updated: 2023-10-10

These services provide information about the integrated file system.

- **IFS_JOB_INFO table function**

The IFS_JOB_INFO table function returns a table that contains information about integrated file system references for a job.

- **IFS_OBJECT_LOCK_INFO table function**

The IFS_OBJECT_LOCK_INFO table function returns a result table that contains a row for each job that is known to be holding a reference, or lock, on the object.

- **IFS_OBJECT_PRIVILEGES table function**

The IFS_OBJECT_PRIVILEGES table function returns a row for every user authorized to the object identified by the path name, along with their associated object and data authorities.

- **IFS_OBJECT_REFERENCES_INFO table function**

The IFS_OBJECT_REFERENCES_INFO table function returns a single row result table that contains information about integrated file system references on an object.

- **IFS_OBJECT_STATISTICS table function**

The IFS_OBJECT_STATISTICS table function returns a table of objects contained in the starting path name or accessible from the starting path name.

- **IFS_READ, IFS_READ_BINARY, and IFS_READ_UTF8 table functions**

The IFS_READ, IFS_READ_BINARY, and IFS_READ_UTF8 table functions read an integrated file system stream file identified by *path-name*. The file's data is returned as character, binary, or UTF-8 data. It can be returned as one string of data, or it can be broken into multiple lines using a specified length or end of line characters.

- **IFS_RENAME scalar function** >

The IFS_RENAME scalar function renames either a file or a directory in the Integrated File System.

- **IFS_WRITE, IFS_WRITE_BINARY, and IFS_WRITE_UTF8 procedures**

The IFS_WRITE, IFS_WRITE_BINARY, and IFS_WRITE_UTF8 procedures write data to an integrated file system stream file. The data can be written as character, binary, or UTF-8 data. Data can be either replaced or appended for an existing file, or a new file can be created.

- **SERVER_SHARE_INFO view**

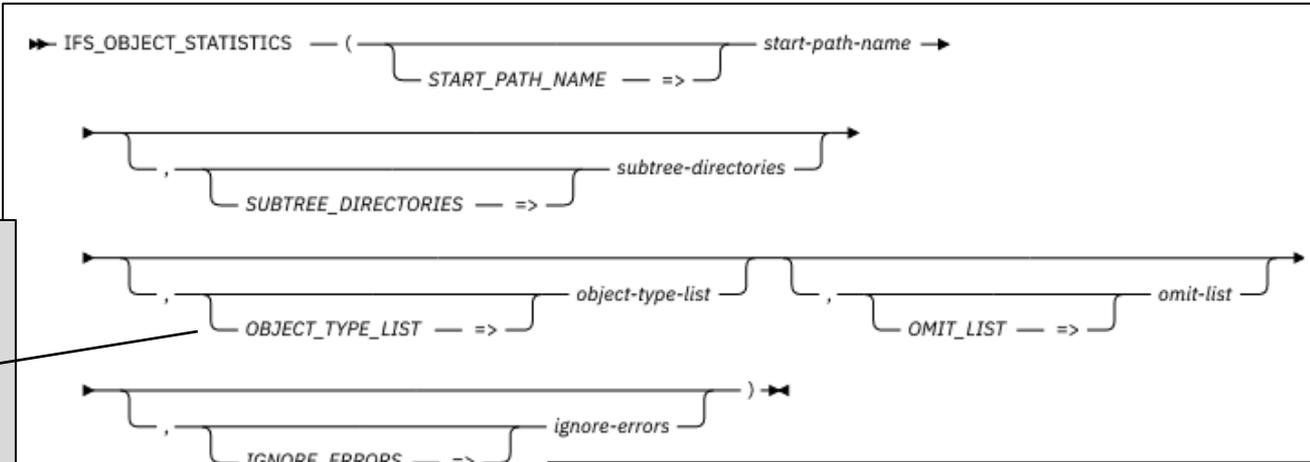
The SERVER_SHARE_INFO view returns information about IBM® i NetServer shares.

Nom	Bib	Type	Description
IFS_OBJECT_STATISTICS	QSYS2	UDTF	Liste et attributs des objets d'un répertoire de l'IFS (~RTVDIRINF)
IFS_OBJECT_PRIVILEGES	QSYS2	UDTF	Droits sur un objet de l'IFS (~DSPAUT)
IFS_OBJECT_LOCK_INFO	QSYS2	UDTF	Travaux qui référencent / verrouillent un objet de l'IFS
IFS_OBJECT_REFERENCES_INFO	QSYS2	UDTF	Références sur un objet de l'IFS
IFS_RENAME	SYSTOOLS	UDF	Renommage d'un répertoire ou d'un fichier de l'IFS
IFS_JOB_INFO	QSYS2	UDTF	Objets IFS référencés par un travail
IFS_READ	QSYS2	UDTF	Lecture d'un fichier IFS – Retour en alpha, CCSID du job
IFS_READ_UTF8	QSYS2	UDTF	Lecture d'un fichier IFS – Retour en Unicode UTF-8
IFS_READ_BINARY	QSYS2	UDTF	Lecture d'un fichier IFS – Retour en binaire
IFS_WRITE	QSYS2	Procédure	Ecriture d'un fichier dans l'IFS – Ecriture en alpha, CCSID du job
IFS_WRITE_UTF8	QSYS2	Procédure	Ecriture d'un fichier dans l'IFS – Ecriture en alpha Unicode
IFS_WRITE_BINARY	QSYS2	Procédure	Ecriture d'un fichier dans l'IFS – Ecriture en binaire
SERVER_SHARE_INFO	QSYS2	Vue	Informations sur les partages Net Server

1. Service IFS_OBJECT_STATISTICS

- Liste et attributs des objets d'un répertoire de l'IFS
- Documentation : <https://www.ibm.com/docs/fr/i/7.4?topic=services-ifs-object-statistics-table-function>

■ Fonction table (QSYS2)
Syntaxe :



- Tous les types d'objet
- + les valeurs spéciales suivantes :
 - *ALLDIR (tous les sous-répertoires)
 - *ALLSTMF (tous les fichiers)
 - *MBR (tous les membres des PF)
 - *NOQDLS (exclure QDLS)
 - *NOQOPT (exclure QOPT et QNTC)
 - *NOQSYS (exclure QSYS)

NO
An error is returned.

YES
A warning is returned.
No row is returned when an error is encountered. This is the default.

- Retourne une table avec 88 colonnes

IFS_OBJECT_STATISTICS

```
33 -- Les objets sous /home - SANS les sous-répertoires - Informations de base
34
35 SELECT Path_Name, Object_Type, Data_Size, Object_Owner
36 FROM TABLE (ifs_object_statistics(START_PATH_NAME => '/home',
37                                     SUBTREE_DIRECTORIES => 'NO'));
38
```

PATH_NAME	OBJECT_TYPE	DATA_SIZE	OBJECT_OWNER
/home	*DIR	8192	QSYS
/home/QSECOFR	*DIR	8192	QSECOFR
/home/GUIBERT	*DIR	8192	QSECOFR
/home/QWQADMIN	*DIR	8192	QWQADMIN
/home/QWQADMGRP	*DIR	8192	QWQADMGRP
/home/QCIUSER	*DIR	8192	QCIUSER
/home/BOURGEOIS	*DIR	8192	QSECOFR
/home/db2mssh	*DIR	8192	QSECOFR
/home/javaTrustStore	*DIR	8192	BOURGEOIS
/home/javaTrustStorejksExport	*STMF	107636	BOURGEOIS
/home/LPAR2RRD	*DIR	8192	QSECOFR
/home/NTR	*DIR	8192	LMERMET
/home/LMERMET	*DIR	12288	LMERMET

IFS_OBJECT_STATISTICS

```
39 -- Les objets sous /home - AVEC les sous-répertoires - Informations de base
40
41 SELECT Path_Name, Object_Type, Data_Size, Object_Owner
42 FROM TABLE (ifs_object_statistics(START_PATH_NAME => '/home',
43 SUBTREE_DIRECTORIES => 'YES')) ORDER BY Path_Name;
44
```

PATH_NAME	OBJECT_TYPE	DATA_SIZE	OBJECT_OWNER
/home	*DIR	8192	QSYS
/home/BOURGEOIS	*DIR	8192	QSECOFR
/home/BOURGEOIS/.bash_history	*STMF	628	BOURGEOIS
/home/BOURGEOIS/.eclipse	*DIR	8192	BOURGEOIS
/home/BOURGEOIS/.eclipse/RSE	*DIR	8192	BOURGEOIS
/home/BOURGEOIS/.iSeriesAccess	*DIR	8192	BOURGEOIS
/home/BOURGEOIS/.iSeriesAccess/cwb_userprefs.ini	*STMF	1076	BOURGEOIS
/home/BOURGEOIS/.isql_history	*STMF	53	BOURGEOIS
/home/BOURGEOIS/.odbc.ini	*STMF	0	BOURGEOIS
/home/BOURGEOIS/.profile	*STMF	99	BOURGEOIS
/home/BOURGEOIS/.psql_history	*STMF	220	BOURGEOIS
/home/BOURGEOIS/.sh_history	*STMF	112	BOURGEOIS
/home/BOURGEOIS/.vscode	*DIR	8192	BOURGEOIS
/home/BOURGEOIS/.vscode/codeforiserver-1.2.0.jar	*STMF	332640	BOURGEOIS
/home/BOURGEOIS/javasharedresources	*DIR	12288	BOURGEOIS
/home/BOURGEOIS/javasharedresources/C290M11...	*STMF	67108864	BOURGEOIS
/home/BOURGEOIS/javasharedresources/C290M11...	*STMF	314572800	BOURGEOIS
/home/BOURGEOIS/logfile	*STMF	979	BOURGEOIS
/home/BOURGEOIS/xsr.log	*STMF	2712	BOURGEOIS
/home/GUIBERT	*DIR	8192	QSECOFR
/home/LMERMET	*DIR	12288	LMERMET
/home/LMERMET/ACMSaveData.tar	*STMF	10240	LMERMET
/home/LMERMET/HMC Update VQR1MQ42 v86 icon	*STMF	4186636788	LMERMET

IFS_OBJECT_STATISTICS

```
50 -- Quels sont les 5 plus gros fichiers sous le répertoire /home ?
51
52 SELECT Path_Name, CHAR(Data_Size/1000000) AS Data_Size_Mo, Last_Used_Timestamp
53 FROM TABLE(ifs_object_statistics(START_PATH_NAME => '/home',
54                                SUBTREE_DIRECTORIES => 'YES',
55                                OBJECT_TYPE_LIST => '*STMF' ))
56 ORDER BY Data_Size DESC LIMIT 5;
```

PATH_NAME	DATA_SIZE_MO	LAST_USED_TIMESTAMP
/home/LMERMET/HMC_Update_V9R1M942_x86.iso	4186	2023-06-15 00:00:00
/home/LMERMET/SaveHSCSystemUpgradeData.tar	1656	2023-03-16 00:00:00
/home/BOURGEOIS/javasharedresources/C290M11F1A64P_sharedcc_bourgeois_...	314	2023-01-05 00:00:00
/home/BOURGEOIS/javasharedresources/C290M11F1A64P_sharedcc_bourgeois_...	67	2022-04-04 00:00:00
/home/NTR/Reports/IBMIV74 NTR 22-12-15T14-35-15-Internal DB-do not open.xml	7	2022-12-15 00:00:00

IFS_OBJECT_STATISTICS

```
43 -- Quels sont les fichiers JSON du répertoire /pb auxquels on a accédé après le 1er octobre 2023 ?
44
45 SELECT Path_Name, Object_Type, Create_Timestamp, Access_Timestamp, Last_Used_Timestamp
46 FROM TABLE(ifs_object_statistics(START_PATH_NAME => '/pb',
47                                SUBTREE_DIRECTORIES => 'YES'))
48 WHERE RIGHT(TRIM(Path_Name), 5) = '.json'
49 AND Access_Timestamp >= '2023-10-01';
50
```

PATH_NAME	OBJECT_TYPE	CREATE_TIMESTAMP	ACCESS_TIMESTAMP	LAST_USED_TIMESTAMP
/pb/json/films.json	*STMF	2022-02-02 12:10:34	2023-10-04 18:23:35	2023-10-04 00:00:00

IFS_OBJECT_STATISTICS

-- Liste des fichiers dans /tmp qui n'ont pas été utilisés depuis au moins 2 ans

```
SELECT Path_Name, Object_Type, Create_Timestamp, Access_Timestamp, Last_Used_Timestamp
FROM TABLE (ifs_object_statistics(START_PATH_NAME => '/tmp',
                                  SUBTREE_DIRECTORIES => 'YES',
                                  OBJECT_TYPE_LIST => '*ALLSTMF'))
WHERE Last_Used_Timestamp < Current Timestamp - 2 years;
```

-- Suppression des fichiers dans /tmp qui n'ont pas été utilisés depuis au moins 2 ans

```
SELECT QCMDEXC('RMVLNK OBJLNK('' CONCAT TRIM(Path_Name) CONCAT '''))
FROM TABLE (ifs_object_statistics(START_PATH_NAME => '/tmp',
                                  SUBTREE_DIRECTORIES => 'YES',
                                  OBJECT_TYPE_LIST => '*ALLSTMF'))
WHERE Last_Used_Timestamp < Current Timestamp - 2 years;
```

- 1 si OK
- -1 si erreur

IFS_OBJECT_STATISTICS

-- Nombre de fichiers et taille globale par répertoire de l'IFS

```
WITH all_ifs_dirs AS
(SELECT Path_Name AS dir_path
 FROM TABLE(ifs_object_statistics(START_PATH_NAME => '/',
                                SUBTREE_DIRECTORIES => 'YES',
                                OBJECT_TYPE_LIST => '*ALLDIR *NOQSYS'))
SELECT dir_path, COUNT(*) AS stmf_count, SUM(data_size) AS total_size FROM
all_ifs_dirs CROSS JOIN TABLE(ifs_object_statistics(START_PATH_NAME => dir_path,
                                                    SUBTREE_DIRECTORIES => 'NO',
                                                    OBJECT_TYPE_LIST => '*STMF'))

GROUP BY dir_path
ORDER BY stmf_count DESC;
```



IFS_OBJECT_STATISTICS

DIR_PATH	STMF_COUNT	TOTAL_SIZE
/QIBM/ProdData/OS/ApplicationServer/runtime/wlp/lib/features/l10n	4481	4176513
/QIBM/ProdData/OS/ApplicationServer/runtime/wlp7/lib/features/l10n	2530	2214275
/QIBM/ProdData/QWEBQRY/ibi/srv77/home/nls	2481	168648539
/QIBM/ProdData/QWEBQRY/ibi/srv77/home/doc	2236	97290735
/QOpenSys/QIBM/UserData/UME/Pegasus/repository/root#cimv2/classes	1778	6343071
/QOpenSys/QIBM/UserData/UME/Pegasus/repository/root#ibmsd/classes	1446	5797421
→ /QOpenSys/pkgs/lib/python2.7/test	1428	23249061
/QOpenSys/pkgs/lib/python3.6/test/__pycache__	1386	28720560
/QIBM/ProdData/OS/ApplicationServer/runtime/wlp/lib	1122	343268912
/QOpenSys/pkgs/lib/postgresql12/share/doc/html	1102	13800000
/QIBM/ProdData/OS/ApplicationServer/runtime/wlp/lib/features	976	1493301
/QIBM/ProdData/QWEBQRY/base80/webapps/webfocus/dhtml/images/qb	732	1324146
/QIBM/ProdData/OS/ApplicationServer/runtime/wlp7/lib	684	159146682
/QIBM/ProdData/QWEBQRY/base80/webapps/webfocus/3rdparty_resourc...	657	14362279
/QIBM/ProdData/HTTPD/admin/wwww	656	2267896

IFS_OBJECT_STATISTICS

-- Les objets de mon répertoire sous /home dont je ne suis pas propriétaire

```
SELECT Path_Name, Object_Type, Object_Owner
FROM TABLE (ifs_object_statistics(START_PATH_NAME => '/home/bourgeois',
                                  SUBTREE_DIRECTORIES => 'YES'))
WHERE Object_Owner <> 'BOURGEOIS' AND Path_Name <> '/home/bourgeois';
```

-- Correction de la problématique

```
SELECT Path_Name,
       QCMDEXC('CHGOWN OBJ('' CONCAT Path_Name CONCAT ''') NEWOWN(BOURGEOIS) RVKOLDAUT(*NO)')
       AS CHGOWN_Result
FROM TABLE (ifs_object_statistics(START_PATH_NAME => '/home/bourgeois',
                                  SUBTREE_DIRECTORIES => 'YES'))
WHERE Object_Owner <> 'BOURGEOIS' AND Path_Name <> '/home/bourgeois';
```

- 1 si OK
- -1 si erreur

IFS_OBJECT_STATISTICS

-- Quels sont les membres dans la bibliothèque PB qui ont été modifiés après le 1er octobre 2023 ?

```
SELECT PATH_NAME,  
       SUBSTR(Path_Name,  
              LOCATE_IN_STRING(Path_Name, '/', 1, 3) + 1,  
              (LOCATE_IN_STRING(Path_Name, '/', 1, 4) - LOCATE_IN_STRING(Path_Name, '/', 1, 3))-1) AS Fichier,  
       SUBSTR(Path_Name, LOCATE_IN_STRING(Path_Name, '/', -1) + 1) AS Membre,  
       Text_Description, Data_Change_Timestamp, Object_Owner  
FROM TABLE(ifs_object_statistics(START_PATH_NAME => '/QSYS.LIB/PB.LIB',  
                                  OBJECT_TYPE_LIST => '*MBR' ))  
WHERE Data_Change_Timestamp >= '2023-09-01'  
ORDER BY Data_Change_Timestamp DESC;
```

PATH_NAME	FICHIER	MEMBRE	TEXT_DESCRIPTION	DATA_CHANGE_TIMESTAMP	OBJECT_OWNER
/QSYS.LIB/PB.LIB/QRPGLESRC.FILE/CONCATARR.MBR	QRPGLESRC.FILE	CONCATARR.MBR	Test de %CONCATARR	2023-10-30 15:21:03	BOURGEOIS
/QSYS.LIB/PB.LIB/QRPGLESRC.FILE/CONCAT.MBR	QRPGLESRC.FILE	CONCAT.MBR	Test de %CONCAT	2023-10-30 15:20:50	BOURGEOIS
/QSYS.LIB/PB.LIB/CLIENTS.FILE/CLIENTS.MBR	CLIENTS.FILE	CLIENTS.MBR	Liste des clients	2023-09-25 19:48:23	PB
/QSYS.LIB/PB.LIB/CINEMAS.FILE/CINEMAS.MBR	CINEMAS.FILE	CINEMAS.MBR	Liste des cinémas de la région parisienne	2023-09-25 17:08:20	PB
/QSYS.LIB/PB.LIB/GEOME00001.FILE/GEOME00001.MBR	GEOME00001.FILE	GEOME00001.MBR	Exemples de données géospaciales	2023-09-15 11:13:34	PB
/QSYS.LIB/PB.LIB/TERCOMMERC.FILE/TERCOMMERC.MBR	TERCOMMERC.FILE	TERCOMMERC.MBR	Territoires commerciaux	2023-09-14 17:33:52	PB
/QSYS.LIB/PB.LIB/EMAILS.FILE/EMAILS.MBR	EMAILS.FILE	EMAILS.MBR	E-mails des contacts IBM i	2023-09-05 16:11:47	BOURGEOIS
/QSYS.LIB/PB.LIB/EMAILS_ERR.FILE/EMAILS_ERR.MBR	EMAILS_ERR.FILE	EMAILS_ERR.MBR	E-mails en erreur des contacts IBM i	2023-09-05 16:11:47	BOURGEOIS
/QSYS.LIB/PB.LIB/EMAILS_SUP.FILE/EMAILS_SUP.MBR	EMAILS_SUP.FILE	EMAILS_SUP.MBR	E-mails à supprimer des contacts IBM i	2023-09-05 16:10:12	BOURGEOIS

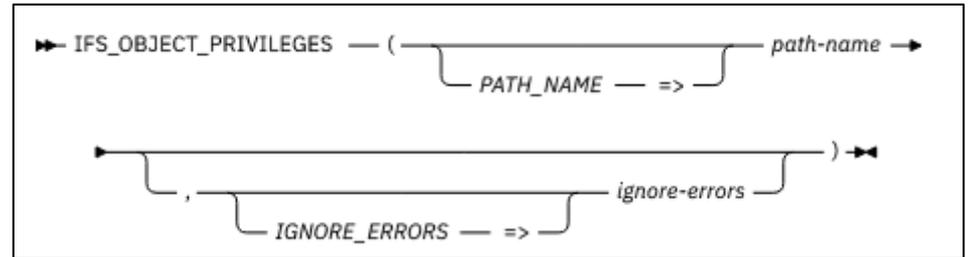
2. Service IFS_OBJECT_PRIVILEGES

~DSPAUT



- Droits sur un objet de l'IFS
- Documentation : <https://www.ibm.com/docs/fr/i/7.4?topic=services-ifs-object-privileges-table-function>

- Fonction table (QSYS2) – Syntaxe :



- Retourne une table avec 17 colonnes

IFS_OBJECT_PRIVILEGES

```
111 -- Quels sont les droits sur le fichier /pb/json/films.json ?
112
113 SELECT * FROM TABLE(ifs_object_privileges('/pb/json/films.json'));
114
```

AUTHORIZATION_NAME	DATA_AUTHORITY	OBJECT_OPERATIONAL	OBJECT_MANAGEMENT	OBJECT_EXISTENCE
*PUBLIC	*RWX	YES	YES	YES
BOURGEOIS	*RWX	YES	YES	YES
QDIRSRV	*X	YES	NO	NO

OBJECT_ALTER	OBJECT_REFERENCE	DATA_READ	DATA_ADD	DATA_UPDATE	DATA_DELETE	DATA_EXECUTE
YES	YES	YES	YES	YES	YES	YES
YES	YES	YES	YES	YES	YES	YES
NO	NO	NO	NO	NO	NO	YES

IFS_OBJECT_PRIVILEGES et IFS_OBJECT_STATISTICS

```

120 -- Objets IFS de /home/bourgeois qui ne sont pas en *PUBLIC *EXCLUDE
121
122 SELECT Path_Name, os.Object_Type, Owner, Data_Authority
123 FROM TABLE(ifs_object_statistics(START_PATH_NAME => '/home/bourgeois',
124 SUBTREE_DIRECTORIES => 'YES')) os,
125 LATERAL (SELECT Object_Type, Owner, Data_Authority
126 FROM TABLE(ifs_object_privileges(PATH_NAME => os.path_name)) op
127 WHERE op.authorization_name = '*PUBLIC' AND data_authority <> '*EXCLUDE');
128

```

PATH_NAME	OBJECT_TYPE	OWNER	DATA_AUTHORITY
/home/bourgeois	*DIR	QSECOFR	*RWX
/home/bourgeois/.eclipse	*DIR	BOURGEOIS	*RWX
/home/bourgeois/.sh_history	*STMF	BOURGEOIS	*NONE
/home/bourgeois/.bash_history	*STMF	BOURGEOIS	*NONE
/home/bourgeois/.odbc.ini	*STMF	BOURGEOIS	*RW
/home/bourgeois/.iSeriesAccess	*DIR	BOURGEOIS	*NONE
/home/bourgeois/.profile	*STMF	BOURGEOIS	*RWX
/home/bourgeois/.isql_history	*STMF	BOURGEOIS	*NONE
/home/bourgeois/.lnqfile	*STMF	BOURGEOIS	*NONE

IFS_OBJECT_PRIVILEGES et IFS_OBJECT_STATISTICS

-- Objets IFS de /home/bourgeois qui ne sont pas en *PUBLIC *EXCLUDE
-- Correction de la problématique

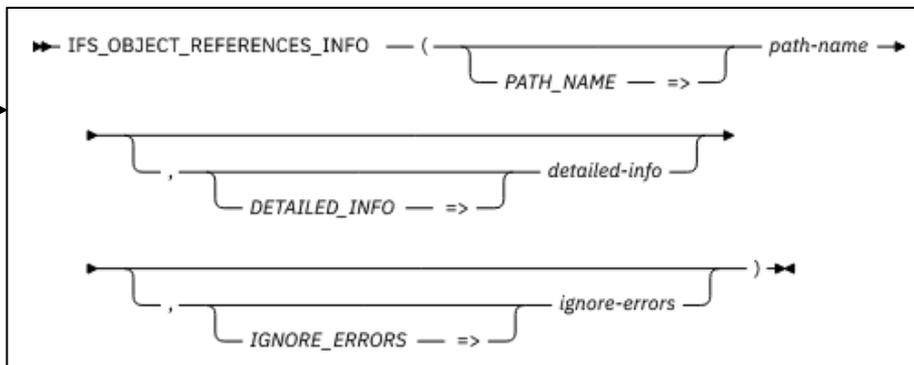
```
SELECT QCMD EXC('CHGAUT OBJ('' CONCAT TRIM(os.Path_Name) CONCAT ''') USER(*PUBLIC) DTAUT(*EXCLUDE) OBJAUT(*NONE)')  
AS CHGAUT_Result  
FROM TABLE(ifs_object_statistics(START_PATH_NAME => '/home/bourgeois',  
SUBTREE_DIRECTORIES => 'YES')) os,  
LATERAL (SELECT Object_Type, Owner, Data_Authority  
FROM TABLE(ifs_object_privileges(PATH_NAME => os.path_name)) op  
WHERE op.authorization_name = '*PUBLIC' AND data_authority <> '*EXCLUDE');
```

PATH_NAME	CHGAUT_RESULT
/home/bourgeois	1
/home/bourgeois/.eclipse	1
/home/bourgeois/.sh_history	1
/home/bourgeois/.bash_history	1
/home/bourgeois/.odbc.ini	1
/home/bourgeois/iSeriesAccess	1

- 1 si OK
- -1 si erreur

3. Service IFS_OBJECT_REFERENCES_INFO

- Nombre de références sur un objet de l'IFS
- Documentation : <https://www.ibm.com/docs/fr/i/7.5?topic=services-ifs-object-references-info-table-function>
- Fonction table (QSYS2)
- Syntaxe :



Retourne une table avec 17 ou 42 colonnes

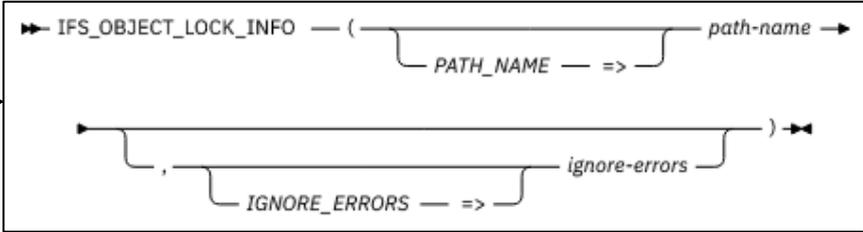
```

220 -- Combien de travaux ont une référence sur le répertoire /home/bourgeois ?
221
222 SELECT * FROM TABLE(ifs_object_references_info(PATH_NAME => '/home/bourgeois'));

```

PATH_NAME	REFERENCE_COUNT	IN_USE	RO_COUNT	WO_COUNT	RW_COUNT	XO_COUNT	SH...
/home/bourgeois	7	YES	0	0	0	0	0

4. Service IFS_OBJECT_LOCK_INFO

- Travaux qui référencent / verrouillent un objet de l'IFS
- Documentation : <https://www.ibm.com/docs/fr/i/7.5?topic=services-ifs-object-lock-info-table-function>
- Fonction table (QSYS2)
- Syntaxe : 

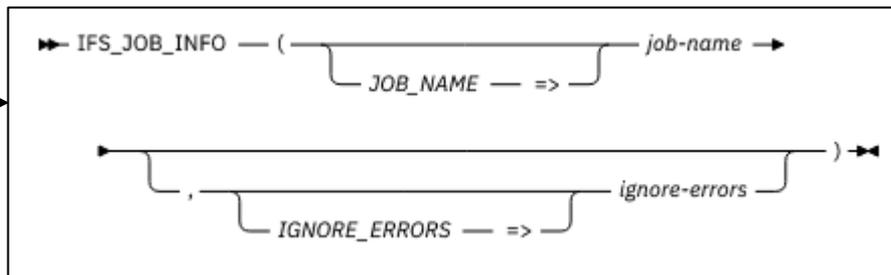
Retourne une table avec 42 colonnes

```
213 -- Quels sont les travaux qui ont un verrouillage ou une référence sur le répertoire /home/bourgeois ?
214
215 SELECT * FROM TABLE(ifs_object_lock_info(PATH_NAME => '/home/bourgeois'));
216
```

PATH_NAME	JOB_NAME	RO_COUNT	WO_COUNT	RW_COUNT	XO_COUNT	SHARE_R_COUNT	SHAR
/home/bourgeois	764187/QUSER/QZDASOINIT	0	0	0	0	0	0
/home/bourgeois	764185/QUSER/QZDASOINIT	0	0	0	0	0	0
/home/bourgeois	764184/QUSER/QZDASOINIT	0	0	0	0	0	0
/home/bourgeois	764156/BOURGEOIS/QPADEV0002	0	0	0	0	0	0
/home/bourgeois	764145/QUSER/QZRCSRVS	0	0	0	0	0	0
/home/bourgeois	763992/QUSER/QZDASOINIT	0	0	0	0	0	0
/home/bourgeois	759191/BOURGEOIS/QB5ROUTER	0	0	0	0	0	0

5. Service IFS_JOB_INFO

- Objets IFS référencés par un travail
- Documentation : <https://www.ibm.com/docs/fr/i/7.4?topic=services-ifs-job-info-table-function>
- Fonction table (QSYS2)
- Syntaxe :



Retourne une table avec 38 colonnes

```

227 -- Quels sont les références IFS de mon travail en cours ?
228
229 SELECT * FROM TABLE(ifs_job_info(JOB_NAME => '*'));
  
```

PATH_NAME	FILE_SYSTEM_TYPE	FILE_IDENTIFIER_NUMBER	GENERATION_IDENTIFIER	FILE_S
/	ROOT	21	2784636197	
/home/BOURGEOIS	ROOT	28015	2784636204	

6. Service **SERVER_SHARE_INFO**

- Informations sur les partages Net Server (fichier et imprimante)
- Documentation : <https://www.ibm.com/docs/fr/i/7.5?topic=services-server-share-info-view>
- Vue (dans QSYS2) – 12 colonnes

```

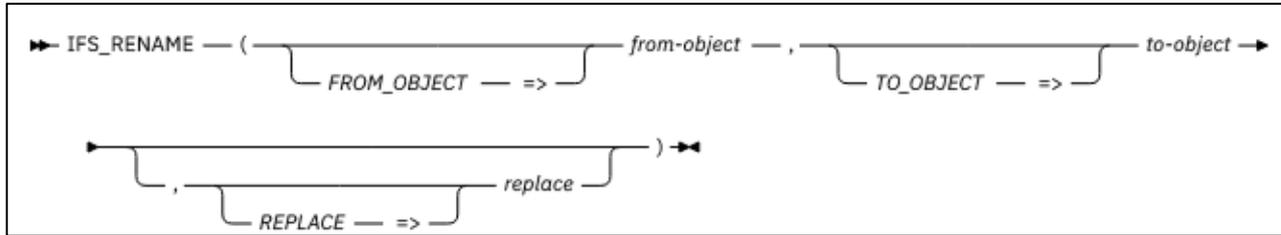
234 -- Quels sont les partages de fichiers Net Server ?
235
236 SELECT * FROM server_share_info
237 WHERE SHARE_TYPE = 'FILE';
---
```

Server Share Name	Share Type	Text Description	Encryption Required	Path Name	Permissions	Maximum Connections	Current Connections	Te Cc En
PB	FILE	Répertoire de PB	NO	/pb	*RW		1	0 NC
PTF	FILE	Images catalogue	NO	/ptf74	*RW		-	1 NC
PHP	FILE	Serveur et fichiers PHP	NO	/phposs	*R		-	0 NC

7. Service IFS_RENAME

Disponible le 17/11/2023

- Permet de renommer un objet de l'IFS
- Documentation : <https://www.ibm.com/docs/fr/i/7.5?topic=services-ifs-rename-scalar-function>
- Fonction scalaire (dans SYSTOOLS) – Syntaxe :



Renvoie 0 si pas d'erreur, sinon le numéro d'erreur

```
VALUES ifs_rename(FROM_OBJECT => '/pb/json/listefilms.json',  
                 TO_OBJECT   => '/pb/json/listefilms' CONCAT  
                               VARCHAR_FORMAT(current date, 'YYYYMMDD')  
                               CONCAT '.json',  
                 REPLACE => 'NO');
```

Lecture et écriture de fichiers dans l'IFS

1. Fonctions SQL de **lecture** de fichiers dans l'IFS

UDF	• GET_BLOB_FROM_FILE	Retourne, en BLOB, le contenu d'un fichier IFS ou membre source
	• GET_CLOB_FROM_FILE	Retourne, en CLOB, le contenu d'un fichier IFS ou membre source
	• GET_DBCLOB_FROM_FILE	Retourne, en DBCLOB, le contenu d'un fichier IFS ou membre source
	• GET_XML_FILE	Retourne, en BLOB, le contenu d'un fichier XML IFS ou membre source
UDTF	• IFS_READ	Retourne, en alphanumérique EBCDIC, le contenu d'un fichier IFS
	• IFS_READ_BINARY	Retourne, en binaire, le contenu d'un fichier IFS
	• IFS_READ_UTF8	Retourne, en alphanumérique UTF-8, le contenu d'un fichier IFS

- **UDF** (User Defined Function) : fonction **scalaire**
- **UDTF** (User Defined Table Function) : fonction **table**

Lecture de fichier IFS – Fonctions `GET_xxx_FROM_FILE`

- Syntaxe : `GET_xxx_FROM_FILE (chemin_du_fichier, <entier>)`
 - Chemin du fichier :
 - Constante ou variable
 - `/rep1/rep2/.../fichier.extension` ou `bibliothèque/fichier_source(membre_source)`
 - Entier
 - Constante égale à 0 ou 1 – Pris en compte pour les membres source uniquement :
 - 0 : les espaces sont renvoyés (valeur par défaut)
 - 1 : les espaces sont supprimés
- Fonctions **scalaires** : renvoient un BLOB, CLOB ou DBCLOB locator → **retournent le contenu du fichier en 1 seule ligne, 1 seule colonne**
 - BLOB : sans conversion du CCSID
 - CLOB : dans le CCSID du job
 - DBCLOB : dans le CCSID DBCS du job
- Doit être exécutée sous **contrôle de validation**
 - Le locator est libéré lors du COMMIT ou ROLLBACK

Lecture de fichier IFS – Fonctions GET_xxx_FROM_FILE

- Exemples

```
films.json 1 {"films":[{"titre":"MEILLEUR (LE)","annee":1984,"duree":129},{"titre":"POURSUITE IMPITOYABLE (LA)","annee":1965,

255 VALUES get_clob_from_file('/pb/json/films.json') WITH CS;

00001
{"films":[{"titre":"MEILLEUR (LE)","annee":1984,"duree":129},{"titre":"POURSUITE IMPITOYABLE (LA)","annee":1965,
```

```
films.properties
#IWS generated property file for service films
#Thu Aug 25 15:19:15 CEST 2022
iws.version=2.0
ws.iws.gen.authenticationmethod=*NONE
```

```
256 VALUES get_clob_from_file('/pb/films.properties') WITH CS;

00001
#IWS generated property file for service films#Thu Aug 25 15:19:15 CEST 2022iws.version=2.0ws.iws.gen.auther
```

Lecture de fichier IFS – Fonctions `GET_xxx_FROM_FILE`

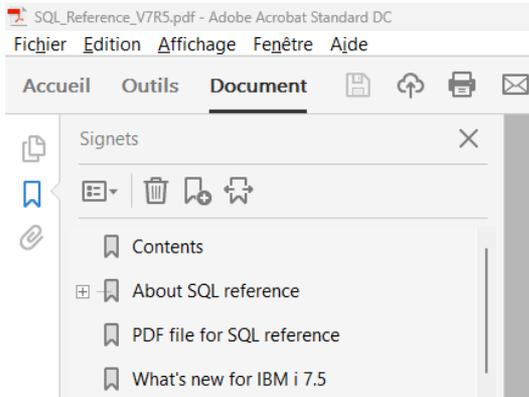
- Exemples

```
films.csv x
pb > csv > films.csv
1 1,12,"MEILLEUR (LE)","O",130,1984
2 29,10,"POURSUITE IMPITOYABLE (LA)","D",135,1965
3 37,6,"RIDEAU DECHIRE (LE)","P",125,1966
```

```
257 VALUES get_clob_from_file('/pb/csv/films.csv') WITH CS;
```

```
00001
```

```
1,12,"MEILLEUR (LE)","O",130,198429,10,"POURSUITE IMPITOYABLE (LA)","D",135,196537,6,"RIDE
```



```
258 VALUES get_blob_from_file('/pb/pdf/SQL_Reference_V7R5.pdf') WITH CS;
```

```
00001
```

```
00001
```

```
255044462D312E360D25E2E3CFD30D0A33343336382030206F626A0D0A3C3C2F4C696E6!
```

Lecture de fichier IFS – Fonctions GET_xxx_FROM_FILE

- Recherche de fichiers IFS

```
260 -- Recherche des fichiers JSON dans /pb qui contiennent "1984"
261
262 SELECT Path_Name, get_clob_from_file(Path_Name) AS "Contenu du fichier IFS"
263 FROM TABLE(ifs_object_statistics(START_PATH_NAME => '/pb',
264                                SUBTREE_DIRECTORIES => 'YES',
265                                OBJECT_TYPE_LIST => '*ALLSTMF'))
266 WHERE RIGHT(Path_Name, 5) = '.json'
267 AND get_clob_from_file(Path_Name) LIKE '%1984%'
268 ORDER BY Path_Name WITH CS;
```

Fonctionnement
identique avec des
membres source

PATH_NAME	Contenu du fichier IFS
/pb/iws/listefilms.json	{"films":[{"titre":"TERMINATOR","annee":1984,"duree":107},{
/pb/json/films.json	{"films":[{"titre":"MEILLEUR (LE)","annee":1984,"duree":129},
/pb/json/listefilms.json	{"films":[{"titre":"MEILLEUR (LE)","annee":1984,"duree":129},

Lecture de fichier IFS – Fonctions GET_xxx_FROM_FILE

- Comparaison de fichiers IFS

```
273 -- Est-ce que les fichiers listefilms.json qui se trouvent dans 2 répertoires différents ont le même contenu ?
274
275 VALUES (CASE
276     WHEN get_clob_from_file('/pb/json/listefilms.json') =
277         get_clob_from_file('/pb/iws/listefilms.json')
278     THEN 'Les 2 fichiers ont le même contenu'
279     ELSE 'Les 2 fichiers ont un contenu différent'
280 END) WITH CS;
281
```

Fonctionnement
identique avec des
membres source

00001

Les 2 fichiers ont un contenu différent

Lecture de fichier IFS – Fonctions GET_xxx_FROM_FILE

- Lecture et décomposition d'un fichier JSON dans l'IFS

```
films.json ⌵  
1 {"films": [{"titre": "MEILLEUR (LE)", "annee": 1984, "duree": 129}, {"titre": "POURSUITE IMPITOYABLE (LA)", "annee": 1965, "duree": 135}]}
```

```
359 -- Lecture et décomposition d'un fichier JSON dans l'IFS  
360  
361 SELECT * FROM JSON_TABLE(  
362   get_clob_from_file('/pb/json/films.json'),  
363   '$.films' COLUMNS(titre VARCHAR(40) PATH '$.titre',  
364                      annee CHAR(4) PATH '$.annee',  
365                      duree CHAR(4) PATH '$.duree')) WITH CS;  
366
```

TITRE	ANNEE	DUREE
MEILLEUR (LE)	1984	129
POURSUITE IMPITOYABLE (LA)	1965	135
DUREE DECHIDE (LE)	1966	125

Lecture de fichier IFS – Fonctions GET_xxx_FROM_FILE

- Lecture et décomposition d'un fichier JSON dans l'IFS

```
318 -- Lecture et décomposition d'un fichier JSON dans l'IFS en passant par une vue
319
320 CREATE OR REPLACE VIEW filmsv1 AS
321 (SELECT * FROM JSON_TABLE(
322   get_clob_from_file('/pb/json/films.json'),
323   '$.films' COLUMNS(titre VARCHAR(40) PATH '$.titre',
324                      annee CHAR(4) PATH '$.annee',
325                      duree CHAR(4) PATH '$.duree')));
326
327
328 SELECT * FROM filmsv1 WITH CS;
```

TITRE	ANNEE	DUREE
MEILLEUR (LE)	1984	129
POURSUITE IMPITOYABLE (LA)	1965	135
DUREAU DECHIDE (LE)	1966	125

Lecture de fichier IFS – Fonction `GET_XML_FILE`

- Syntaxe : `GET_XML_FILE (chemin_du_fichier_XML)`
 - Chemin du fichier XML :
 - Constante ou variable
 - `/rep1/rep2/.../fichier.xml` ou `bibliothèque/fichier_source(membre_source)`

- Renvoie un BLOB locator → retourne le contenu du fichier en 1 seule ligne, 1 seule colonne

- Convertit les données en UTF-8

- Doit être exécutée sous **contrôle de validation**
 - Le locator est libéré lors du COMMIT ou ROLLBACK

Lecture de fichier IFS – Fonction `GET_XML_FILE`

- Exemples

```
320 VALUES get_xml_file('pb/xml/films.xml') WITH CS;
```

```
00001
```

```
3C3F786D6C2076657273696F6E3D22312E302220656E636F64696E673D2255
```

```
322 VALUES XMLPARSE(DOCUMENT get_xml_file('pb/xml/films.xml')) WITH CS;
```

```
323
```

```
00001
```

```
<films><film><titre>ANGEL HEART</titre><annee>1987</annee><duree>115</duree></film><film><titre>A
```

Lecture de fichier IFS – Fonction GET_XML_FILE

- Exemples

```

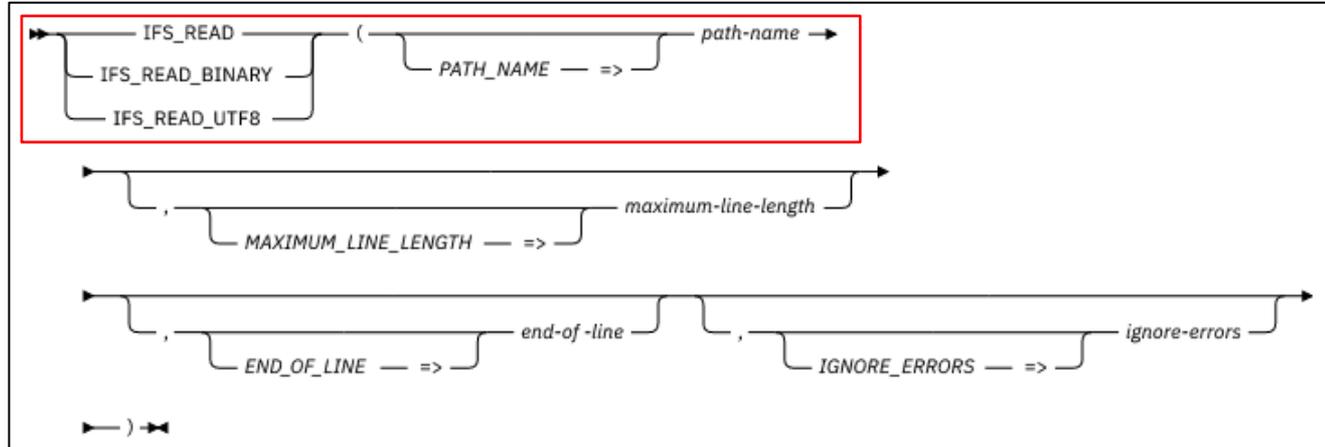
338 SELECT * FROM XMLTABLE('/films/film'
339   PASSING XMLPARSE(DOCUMENT get_xml_file('/pb/xml/films.xml'))
340   COLUMNS titre    VARCHAR(40) PATH 'titre',
341             annee   CHAR(4)      PATH 'annee',
342             duree   CHAR(3)      PATH 'duree') WITH CS;
343

```

TITRE	ANNEE	DUREE
ANGEL HEART	1987	115
APPRENTIS (LES)	1995	95
BANDOLERO !	1968	115
BONHEUR EST DANS LE PRE (LE)	1995	106
CHACUN CHERCHE SON CHAT	1995	95
CHOIX DES ARMES (LE)	1981	140
COMLOT DE FAMILLE	1976	125

Lecture de fichier IFS – Services IFS_READ_XXX

- Syntaxe →



- Fonctions table (UDTF)**

Les données sont renvoyées en une seule chaîne de caractères ou peuvent être scindées en plusieurs lignes (en fonction de la valeur du paramètre END_OF_LINE)

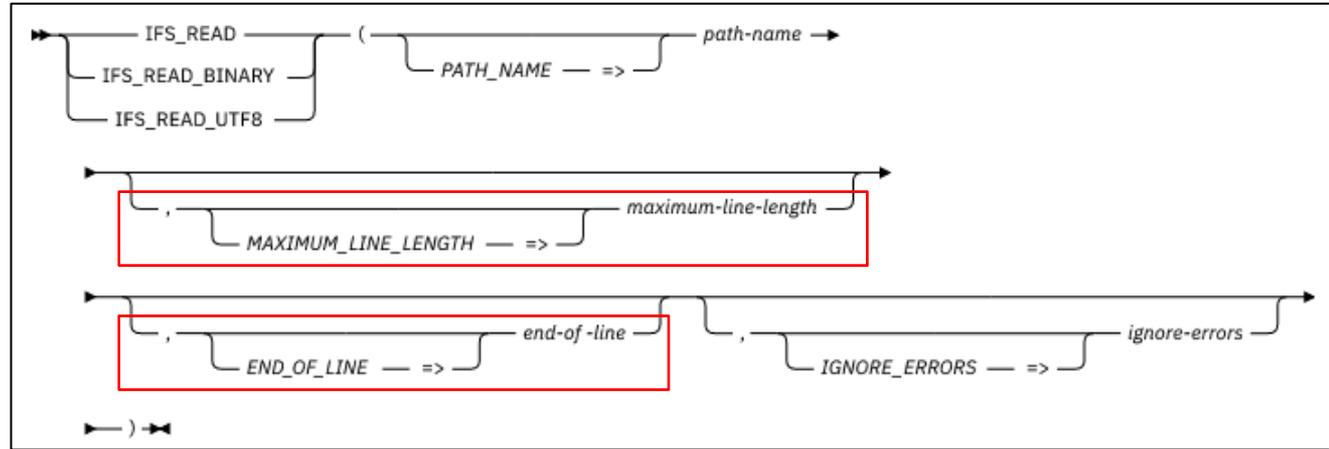
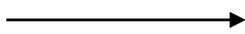
- Trois fonctions table :

- **IFS_READ** Les données renvoyées sont converties en alphanumérique **EBCDIC**
- **IFS_READ_BINARY** Les données renvoyées **ne sont pas converties**
- **IFS_READ_UTF8** Les données renvoyées sont converties en **UTF-8**

- **PATH_NAME** : chemin et nom du fichier (constante ou variable)
(/rep1/rep2/.../fichier.extension)

Lecture de fichier IFS – Services IFS_READ_xxx

- Syntaxe



- MAXIMUM_LINE_LENGTH** Nombre maximum de caractères par ligne (2 Go par défaut)
 - END_OF_LINE** Le ou les caractères qui permettent d'indiquer la fin de ligne
 - Valeurs spéciales :
 - ANY** N'importe quel caractère entre CR, CRLF, LF et LFCR (valeur par défaut pour IFS_READ et IFS_READ_UTF8)
 - CR, CRLF, LF, LFCR** Retour chariot (CR – Carriage Return) / Saut de ligne (LF – Line Feed)
 - NONE** Pas de caractère de fin de ligne (valeur par défaut pour IFS_READ_BINARY)

Lecture de fichier IFS – Services **IFS_READ_xxx**

- Fonctions **table** : retournent une table avec 2 colonnes :
 - **LINE_NUMBER**
 - Pour les 3 fonctions **INTEGER** Position relative de la ligne dans le fichier IFS
 - **LINE**
 - Si **IFS_READ** **CLOB (2 Go)** Données du fichier, dans le CCSID du travail
 - Si **IFS_READ_BINARY** **BLOB (2 Go)** Données du fichier, sans conversion
 - Si **IFS_READ_UTF8** **CLOB (2 Go)** Données du fichier, dans le CCSID 1208 (UTF-8)

- Ne nécessitent pas d'être sous contrôle de validation

Lecture de fichier IFS – Services IFS_READ_xxx

■ Exemples

```
films.json ⌵
1 {"films":[{"titre":"MEILLEUR (LE)","annee":1984,"duree":129},{"titre":"POURSUITE IMPITOYABLE (LA)","annee":1965,"duree":110}]}
```

```
351 SELECT * FROM TABLE(ifs_read(PATH_NAME => '/pb/json/films.json',
352                               END_OF_LINE => 'NONE'));
```

LINE_NUMBER	LINE
1	{"films":[{"titre":"MEILLEUR (LE)","annee":1984,"duree":129},{"titre":"POURSUITE IMPITOYABLE (LA)","annee":1965,"duree":110}]}

```
346 SELECT * FROM TABLE(ifs_read(PATH_NAME => '/pb/films.properties',
347                               END_OF_LINE => 'ANY'));
```

LINE_NUMBER	LINE
1	#IWS generated property file for service films
2	#Thu Aug 25 15:19:15 CEST 2022
3	iws.version=2.0
4	ws.iws.gen.authenticationmethod=*NONE
5	ws.iws.gen.autostartup=true
6	ws.iws.gen.defaultschema=*LIBL

```
films.properties
#IWS generated property file for service films
#Thu Aug 25 15:19:15 CEST 2022
iws.version=2.0
ws.iws.gen.authenticationmethod=*NONE
ws.iws.gen.autostartup=true
ws.iws.gen.defaultschema=*LIBL
```

Lecture de fichier IFS – Services IFS_READ_xxx

- Exemple

```

355 SELECT * FROM JSON_TABLE(
356   (SELECT LINE FROM TABLE(ifs_read(PATH_NAME => '/pb/json/films.json',
357                                END_OF_LINE => 'NONE'))),
358   '$.films' COLUMNS(titre VARCHAR(40) PATH '$.titre',
359                      annee CHAR(4) PATH '$.annee',
360                      duree CHAR(4) PATH '$.duree'));
361

```

TITRE	ANNEE	DUREE
MEILLEUR (LE)	1984	129
POURSUITE IMPITOYABLE (LA)	1965	135
RIDEAU DECHIRE (LE)	1966	125
HOMMES DU PRESIDENT (LES)	1976	126
PSYCHOSE	1960	110
GOOD MORNING, VIETNAM !	1988	117
HOMME DE L'OUEST (L')	1958	94

Lecture de fichier IFS – Services IFS_READ_xxx

■ Recherche de fichiers IFS

```
373 -- Recherche des fichiers JSON dans /pb qui contiennent "1984"
374
375 SELECT Path_Name, (SELECT LINE
376                     FROM TABLE(ifs_read(PATH_NAME => Path_Name,
377                                         END_OF_LINE => 'NONE')) AS "Contenu du fichier IFS"
378 FROM TABLE(ifs_object_statistics(START_PATH_NAME => '/pb',
379                                  SUBTREE_DIRECTORIES => 'YES',
380                                  OBJECT_TYPE_LIST => '*ALLSTMF'))
381 WHERE RIGHT(Path_Name, 5) = '.json'
382 AND (SELECT LINE FROM TABLE(ifs_read(PATH_NAME => Path_Name,
383                                  END_OF_LINE => 'NONE')) LIKE '%1984%'
384 ORDER BY Path_Name;
```

Alternative à l'utilisation
de `get_clob_from_file`
(cf page 37)

PATH_NAME	Contenu du fichier IFS
/pb/iws/listefilms.json	{"films":[{"titre":"TERMINATOR","annee":1984,"duree":107},{"titre":"POURSUITE...
/pb/json/films.json	{"films":[{"titre":"MEILLEUR (LE)","annee":1984,"duree":129},{"titre":"POURSUIT...
/pb/json/films2.json	{"films":[{"titre":"MEILLEUR (LE)","annee":1984,"duree":129},{"titre":"POURSUIT...
/pb/json/listefilms.json	{"films":[{"titre":"MEILLEUR (LE)","annee":1984,"duree":129},{"titre":"POURSUIT...

Lecture de fichier IFS – Services IFS_READ_xxx

- Exemple

```
films.csv x
pb > csv > films.csv
1 1,12,"MEILLEUR (LE)","O",130,1984
2 29,10,"POURSUITE IMPITOYABLE (LA)","D",135,1965
3 37,6,"RIDEAU DECHIRE (LE)","P",125,1966
```

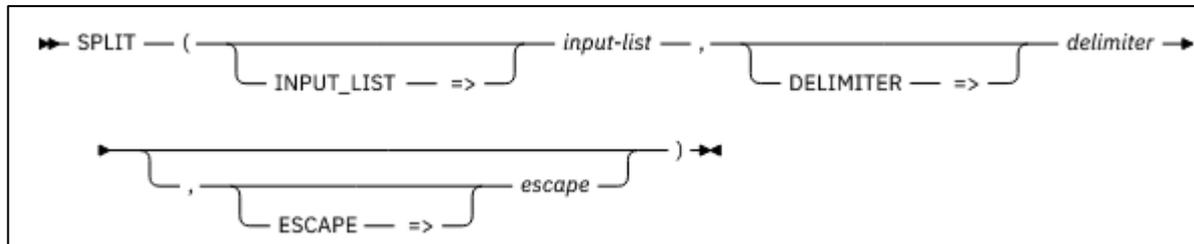
```
352 SELECT LINE FROM TABLE(ifs_read_utf8(PATH_NAME => '/pb/csv/films.csv',
353                                     END_OF_LINE => 'ANY'));
354
```

LINE
1,12,"MEILLEUR (LE)","O",130,1984
29,10,"POURSUITE IMPITOYABLE (LA)","D",135,1965
37,6,"RIDEAU DECHIRE (LE)","P",125,1966
64,2,"HOMMES DU PRESIDENT (LES)","H",126,1976

La fonction table **SPLIT** de SYSTOOLS

- Permet de décomposer en lignes des éléments d'une liste séparés par un délimiteur

- Syntaxe →



- **INPUT_LIST** Chaîne contenant la liste des éléments à décomposer. Les éléments doivent être séparés par le délimiteur
- **DELIMITER** Chaîne contenant le séparateur
- Fonction **table** de **SYSTOOLS** – Retourne une table avec 2 colonnes :
 - **ORDINAL_POSITION** INTEGER Position relative de l'élément dans la chaîne à décomposer
 - **ELEMENT** CLOB (2 Go) Valeur de l'élément

La fonction table **SPLIT** de SYSTOOLS

- Exemples :

```
367 SELECT *
368 FROM TABLE(systools.split('A,B,C,D,E,F,G,H', ','));
```

ORDINAL_POSITION	ELEMENT
1	A
2	B
3	C
4	D
5	E
6	F
7	G
8	H

```
370 WITH cte1 (z1) AS (VALUES('A,B,C,D,E'), ('F,G,H'))
371 SELECT ordinal_position, z1, element
372 FROM cte1 CROSS JOIN TABLE(systools.split(z1, ','));
373 ORDER BY z1, ordinal_position;
```

ORDINAL_POSITION	Z1	ELEMENT
1	A,B,C,D,E	A
2	A,B,C,D,E	B
3	A,B,C,D,E	C
4	A,B,C,D,E	D
5	A,B,C,D,E	E
1	F,G,H	F
2	F,G,H	G
3	F,G,H	H

Lecture de fichier IFS – Services IFS_READ_xxx

- Exemple

```
films.csv x
pb > csv > films.csv
1 1,12,"MEILLEUR (LE)","O",130,1984
2 29,10,"POURSUITE IMPITOYABLE (LA)","D",135,1965
3 37,6,"RIDEAU DECHIRE (LE)","P",125,1966
```

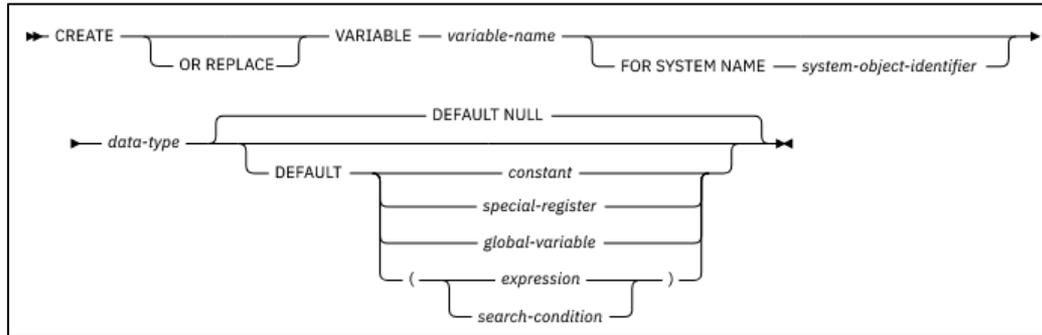
```
376 -- Lecture et décomposition d'un fichier CSV de l'IFS
377
378 WITH cte1 AS (SELECT line_number, line FROM TABLE(ifs_read_utf8(PATH_NAME => '/pb/csv/films.csv'))),
379      cte2 AS (SELECT line_number, line, ordinal_position, TRIM(B "" FROM element) AS element
380              FROM cte1 CROSS JOIN TABLE(systools.split(line, ',')))
381 SELECT line_number,
382        MIN(CASE WHEN ordinal_position = 1 THEN element END) AS code_film,
383        MIN(CASE WHEN ordinal_position = 2 THEN element END) AS code_realisateur,
384        MIN(CASE WHEN ordinal_position = 3 THEN element END) AS titre_film,
385        MIN(CASE WHEN ordinal_position = 4 THEN element END) AS genre_film,
386        MIN(CASE WHEN ordinal_position = 5 THEN element END) AS duree_film,
387        MIN(CASE WHEN ordinal_position = 6 THEN element END) AS annee_film
388 FROM cte2
389 GROUP BY line_number;
```

LINE_NUMBER	CODE_FILM	CODE_REALISATEUR	TITRE_FILM	GENRE_FILM	DUREE_FILM	ANNEE_FILM
11	12		MEILLEUR (LE)	O	130	1984
229	10		POURSUITE IMPITOYABLE (LA)	D	135	1965
337	6		RIDEAU DECHIRE (LE)	P	125	1966
464	7		HOMMES DU PRESIDENT (LES)	H	126	1976

Variables globales SQL

- Objets permanents (dans un schéma/bibliothèque) créés par l'instruction SQL **CREATE VARIABLE**
- Peut-être de tout type SQL (sauf ROWID et DATALINK)
- Le contenu est géré par job / connexion
 - La même variable peut prendre différentes valeurs dans différents jobs / connexions

Syntaxe →



- Si non qualifiée par le nom du schéma :
 - Création : créée dans le CURRENT SCHEMA
 - Utilisation : recherche dans le CURRENT PATH
- Peut être utilisée dans les définitions des vues et des triggers

Lecture de fichier IFS – Services IFS_READ_xxx

```
CREATE OR REPLACE VARIABLE pb.fichier_ifs VARCHAR(1024);

CREATE OR REPLACE VIEW pb.filmsv2 AS
WITH cte1 AS (SELECT line_number, line FROM TABLE(ifs_read_utf8(PATH_NAME => pb.fichier_ifs))),
cte2 AS (SELECT line_number, line, ordinal_position, TRIM(B '' FROM element) AS element
FROM cte1 CROSS JOIN TABLE(systools.split(line, ',')))
SELECT line_number,
MIN(CASE WHEN ordinal_position = 1 THEN element END) AS code_film,
MIN(CASE WHEN ordinal_position = 2 THEN element END) AS code_realisateur,
MIN(CASE WHEN ordinal_position = 3 THEN element END) AS titre_film,
MIN(CASE WHEN ordinal_position = 4 THEN element END) AS genre_film,
MIN(CASE WHEN ordinal_position = 5 THEN element END) AS duree_film,
MIN(CASE WHEN ordinal_position = 6 THEN element END) AS annee_film
FROM cte2
GROUP BY line_number;
```

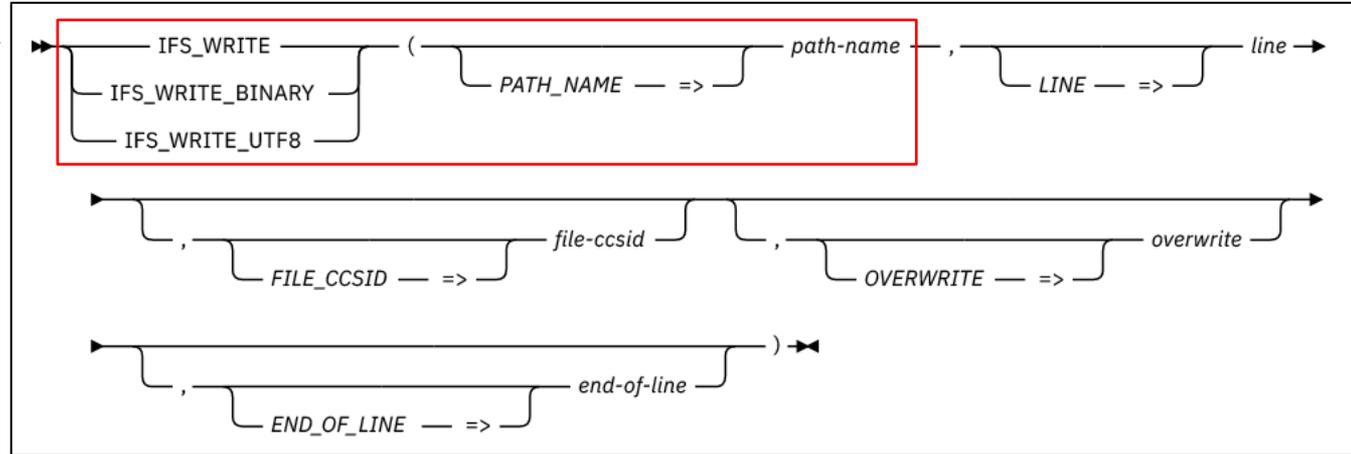
Avec utilisation d'une variable globale et d'une vue

```
407 SET pb.fichier_ifs = '/pb/csv/films.csv';
408
409 SELECT * FROM pb.filmsv2;
410
```

LINE_NUMBER	CODE_FILM	CODE_REALISATEUR	TITRE_FILM	GENRE_FILM	DUREE_FILM	A
11	12		MEILLEUR (LE)	O	130	19
229	10		POURSUITE IMPITOYABLE (LA)	D	135	19
227	6		IDEAL DECHIRE (LE)	D	125	19

Ecriture de fichier IFS – Services IFS_WRITE_XXX

■ Syntaxe



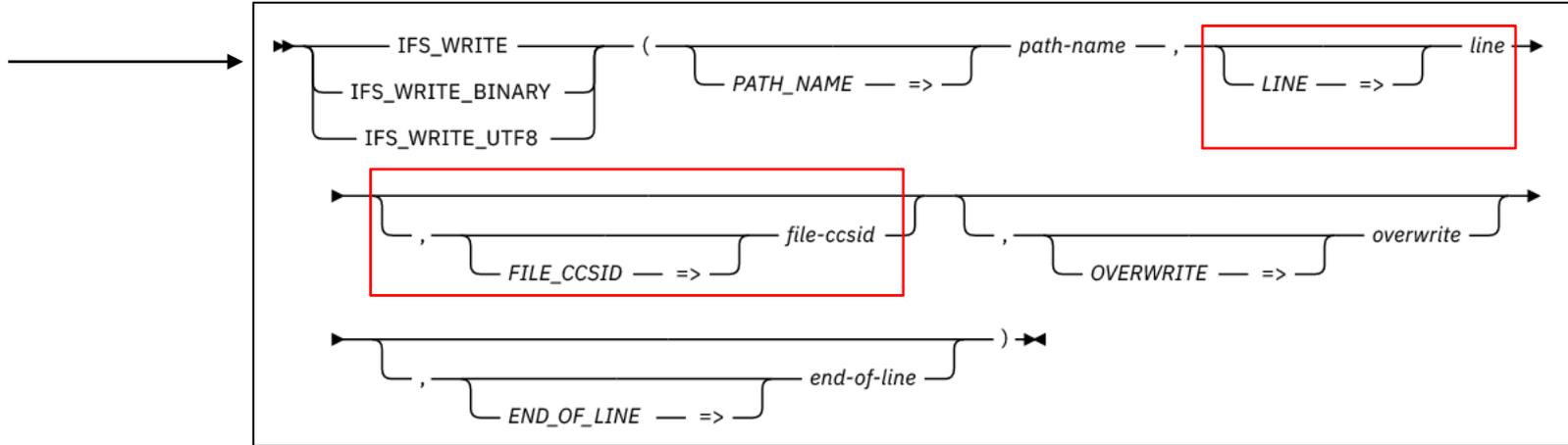
■ Trois procédures :

- **IFS_WRITE** Les données écrites sont converties en alphanumérique **EBCDIC**
- **IFS_WRITE_BINARY** Les données écrites **ne sont pas converties**
- **IFS_WRITE_UTF8** Les données écrites sont converties en **UTF-8**

■ **PATH_NAME** : constante ou variable (/rep1/rep2/.../fichier.extension)

Ecriture de fichier IFS – Services IFS_WRITE_xxx

Syntaxe



LINE

- IFS_WRITE
- IFS_WRITE_BINARY
- IFS_WRITE_UTF8

Les données à écrire

- Une chaîne alphanumérique dans le CCSID du travail
- Une chaîne en binaire
- Une chaîne en UTF-8

FILE_CCSID

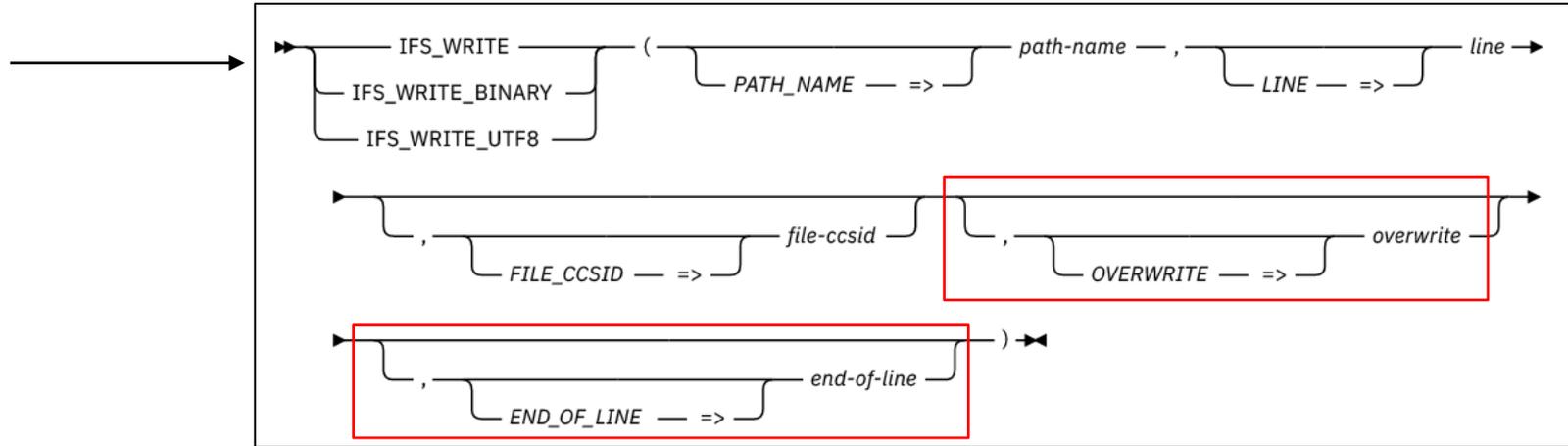
- IFS_WRITE_UTF8
- IFS_WRITE
- IFS_WRITE_BINARY

Le CCSID du fichier cible (si le fichier n'existe pas déjà)

- Valeur par défaut : 1208
 - Valeur par défaut : 0
 - Valeur par défaut : 0
- Si FILE_CCSID = 0, le fichier sera créé avec le CCSID du travail

Ecriture de fichier IFS – Services IFS_WRITE_xxx

Syntaxe



OVERWRITE

Attribut de substitution

- APPEND - Si le fichier existe déjà, les données sont ajoutées à la fin. S'il n'existe pas, il est créé. Valeur par défaut
- NONE - Si le fichier existe déjà, erreur. S'il n'existe pas, il est créé
- REPLACE - Si le fichier existe déjà, il est remplacé. S'il n'existe pas, il est créé

END_OF_LINE

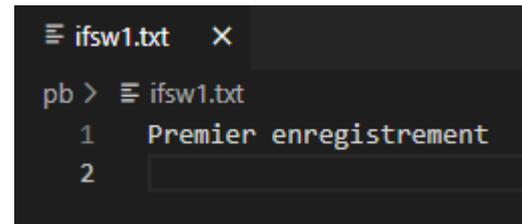
Ajout de caractères de fin de ligne

- CR, CRLF, LF, LFCR - Retour chariot (CR – Carriage Return) / Saut de ligne (LF – Line Feed)
 - La valeur par défaut est CRLF pour IFS_WRITE et IFS_WRITE_UTF8)
- NONE - Pas de caractère de fin de ligne (valeur par défaut et pour IFS_WRITE_BINARY)

Ecriture de fichier IFS – Services IFS_WRITE_XXX

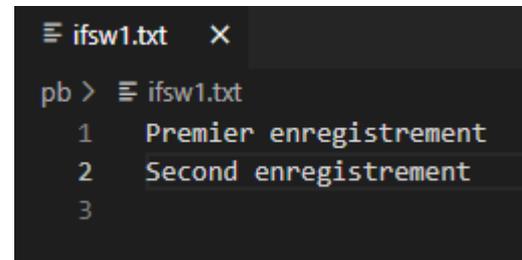
- Exemple

```
CALL ifs_write_utf8(PATH_NAME => '/pb/ifsw1.txt',  
                    LINE => 'Premier enregistrement',  
                    FILE_CCSD => 1208,  
                    OVERWRITE => 'REPLACE');
```



```
ifsw1.txt x  
pb > ifsw1.txt  
1 Premier enregistrement  
2
```

```
CALL ifs_write_utf8(PATH_NAME => '/pb/ifsw1.txt',  
                    LINE => 'Second enregistrement',  
                    OVERWRITE => 'APPEND');
```



```
ifsw1.txt x  
pb > ifsw1.txt  
1 Premier enregistrement  
2 Second enregistrement  
3
```

Ecriture de fichier IFS – Services IFS_WRITE_XXX

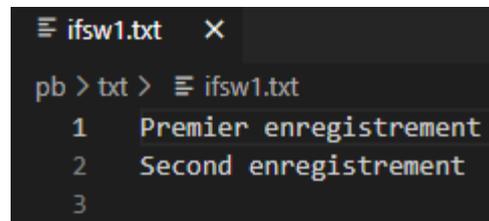
- Exemple

-- Copie d'un fichier IFS d'un répertoire à un autre

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
```

```
CALL ifs_write_utf8(PATH_NAME => '/pb/txt/ifsw1.txt',  
                    LINE => get_clob_from_file('/pb/ifsw1.txt'),  
                    OVERWRITE => 'REPLACE');
```

```
COMMIT;
```



```
ifsw1.txt X  
pb > txt > ifsw1.txt  
1 Premier enregistrement  
2 Second enregistrement  
3
```

Le répertoire cible doit exister

Ecriture de fichier IFS – Services IFS_WRITE_xxx

- Exemple

-- Création d'un fichier JSON dans l'IFS à partir de données d'une table

```
CALL ifs_write_utf8(PATH_NAME => '/pb/json/films2.json',  
                   OVERWRITE => 'REPLACE',  
                   LINE => (SELECT JSON_OBJECT('films':JSON_ARRAYAGG(JSON_OBJECT('titre' VALUE RTRIM(titre),  
                                                                                   'annee' VALUE annee,  
                                                                                   'duree' VALUE duree)))  
                               FROM video.films));
```

```
{ } films2.json X
```

```
pb > json > { } films2.json > ...
```

```
1 {"films":[{"titre":"MEILLEUR (LE)","annee":1984,"duree":129}, {"titre":"POURSUITE IMPITOYABLE (LA)","annee":1965,"duree":135}, {"t
```

Ecriture de fichier IFS – Services IFS_WRITE_XXX

■ Exemple

```
BEGIN
CALL ifs_write_utf8(PATH_NAME => '/pb/rep_ifs.txt',
                    LINE => "",
                    OVERWRITE => 'REPLACE',
                    END_OF_LINE => 'NONE');

FOR SELECT path_name AS nom_repertoire
FROM TABLE(ifs_object_statistics(START_PATH_NAME => '/',
                                SUBTREE_DIRECTORIES => 'NO',
                                OBJECT_TYPE_LIST => '*ALLDIR')) DO

CALL ifs_write_utf8(PATH_NAME => '/pb/rep_ifs.txt',
                    LINE => nom_repertoire);

END FOR;
END;
```

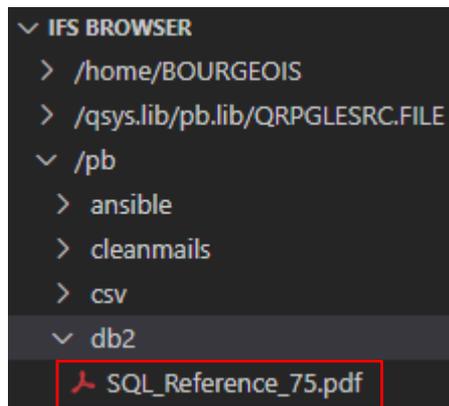
```
rep_ifs.txt x
pb > rep_ifs.txt
1 /
2 /QOpenSys
3 /QDLS
4 /QSYS.LIB
5 /QOPT
6 /QNTC
7 /dev
8 /home
9 /tmp
10 /etc
11 /usr
12 /QIBM
13 /QSR
14 /bin.prv
15 /var
16 /www
17 /QTCPTMM
18 /imgclg
19 /log
20 /run
21 /backups
22 /pb
23 /.iSeriesAccess
24 /temp
25 /GWEBSERP
26 /opt
27 /ptf74
28
```

Ecriture de fichier IFS – Services IFS_WRITE_xxx

- Exemple

-- Ajout dans l'IFS du PDF "SQL Reference" qui se trouve sur le site Web Infocenter d'IBM

```
CALL ifs_write_binary(PATH_NAME => '/pb/db2/SQL_Reference_75.pdf',  
                     OVERWRITE => 'REPLACE',  
                     LINE => HTTP_GET_BLOB('https://www.ibm.com/docs/en/ssw_ibm_i_75/pdf/rbafzpdf.pdf'));
```





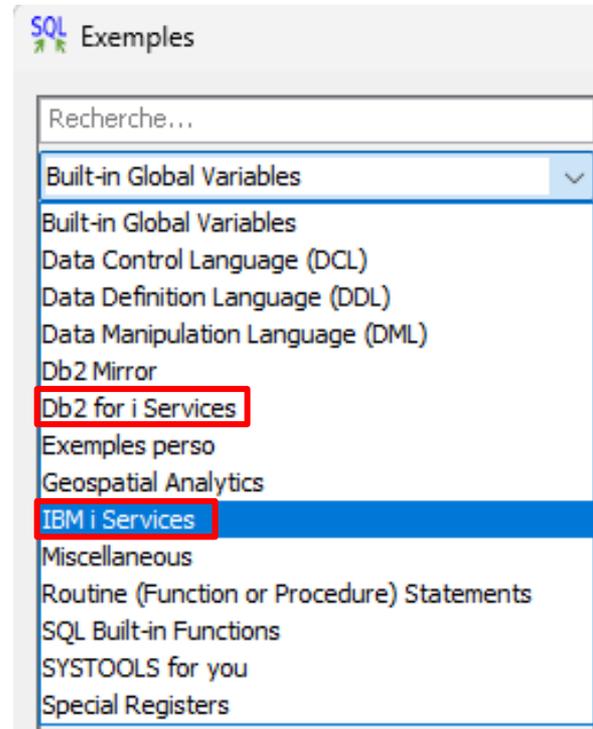
IBM i



3. Pour aller plus loin

Services IBM i / DB2 – Où trouver des exemples ?

- 1. Dans **ACS** – Exécution de scripts SQL



Services IBM i / DB2 – Où trouver des exemples ?

- 1. Dans **ACS** – Exécution de scripts SQL

The screenshot shows the 'SQL Exemples' dialog box. On the left, a search bar is at the top, followed by a dropdown menu showing 'IBM i Services'. Below this is a list of example titles, with 'IFS - 10 largest files under a subdir and tree' selected. The main area contains a SQL script:

```
-- category: IBM i Services
-- description: IFS - 10 largest files under a subdir and tree
-- minvrm: V7R3M0
--
select path_name, object_type, data_size, object_owner
  from table(qsys2.IFS_OBJECT_STATISTICS (
           start_path_name => '/usr',
           subtree_directories => 'YES'))
 order by 3 desc
 limit 10;
```

Overlaid on the bottom right is the 'SQL Préférences' dialog. It contains two checked checkboxes:

- Tri des catégories
- Masquage des exemples qui ne s'appliquent pas au niveau d'édition de système en cours (si connecté)

 At the bottom of the main dialog, the 'Insertion' button is circled in red.

Services IBM i / DB2 – Où trouver des exemples ?

- 2. Dans l'assistant de création de Services Web avec IWS

Deploy New Service
Specify SQL Statements - Step 4 of 8

Specify SQL statements that will be externalized as a Web service: ?

	Procedure name	SQL statement/Parameter name
<input checked="" type="checkbox"/>	<input type="text"/>	

Services IBM i / DB2 – Où trouver des exemples ?

- 2. Dans l'assistant de création de Services Web avec IWS

Insert SQL statement from example

SQL categories: Data Definition Language (DDL) ▾

SQL descriptions: Create Schema ▾

IBM i Services ▾

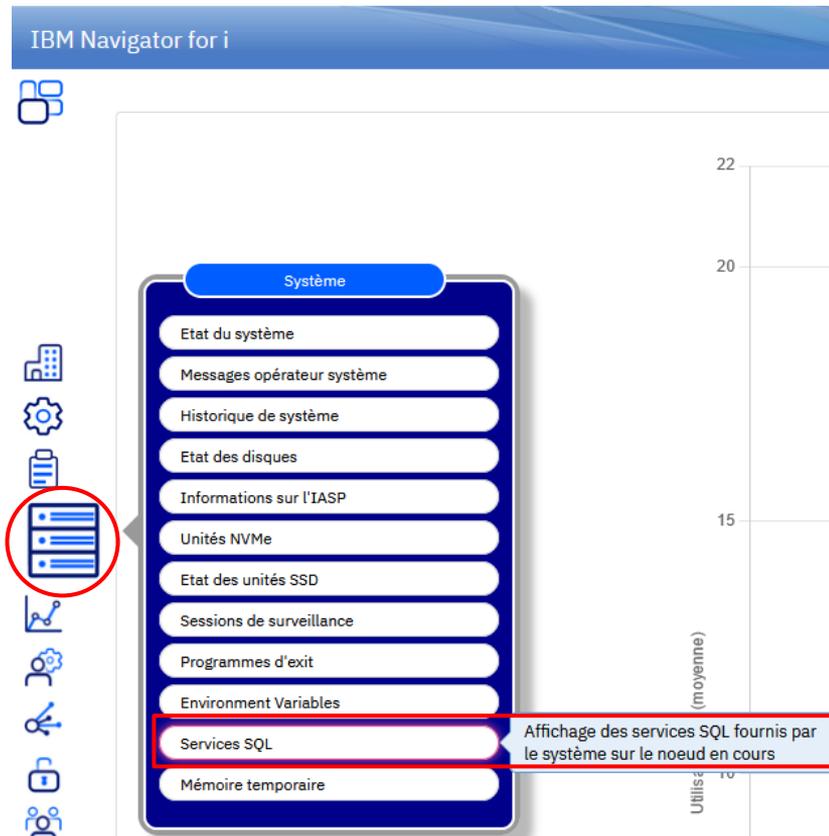
- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Control Language (DCL)
- Routine (Function or Procedure) Statements
- Special Registers
- Built-in Global Variables
- Miscellaneous
- IBM i Services**
- DB2 for i Services

Security - Review *ALLOBJ users

- Security - Review *JOBCTL users
- Security - User Info Sign On Failures
- Security - Group profile detail
- Security - Authorization List detail
- Security - User Info close to disabled
- Security - DRDA Authentication Entry info
- Security - DISPLAY_JOURNAL() of the audit journal
- Security - Which users are changing data via a remote connection?

Services IBM i / DB2 – Où trouver des exemples ?

- 3. Dans IBM Navigator for i



Services IBM i / DB2 – Où trouver des exemples ?

■ 3. Dans IBM Navigator for i

Services SQL IBM i

Catégorie	Nom de schéma	Nom	Type d'objet SQL	Type d'objet	Nom d'objet système	Dernier niveau de groupe Db2	Niveau de groupe Db2 initial	Édition la plus ancienne possible	Exemple
DATABASE-UTILITY	QSYS2	END_IDLE_SQE_THREADS	PROCEDURE			13	13	V7R3M0	-- Description: End th complémentaires<<
DATABASE-UTILITY	QSYS2	SYSFILES	VIEW	*FILE	SYSFILES	25	15	V7R3M0	-- Description: Check complémentaires<<
IFS	QSYS2	IFS_OBJECT_STATISTICS	TABLE FUNCTION			13	4	V7R3M0	-- Description: List ba complémentaires<<
IFS	QSYS2	IFS_JOB_INFO	TABLE FUNCTION			4	4	V7R3M0	-- Description: List all complémentaires<<
IFS	QSYS2	IFS_OBJECT_LOCK_INFO	TABLE FUNCTION			4	4	V7R3M0	-- Description: List all complémentaires<<
IFS	QSYS2	IFS_OBJECT_REFERENCES_INFO	TABLE FUNCTION			4	4	V7R3M0	-- Description: Look a complémentaires<<
IFS	QSYS2	IFS_OBJECT_PRIVILEGES	TABLE FUNCTION			7	7	V7R3M0	-- Description: List all complémentaires<<
IFS	QSYS2	SERVER_SHARE_INFO	VIEW	*FILE	SHARE_INFO	10	10	V7R3M0	-- Description: List all complémentaires<<
IFS	QSYS2	IFS_READ	TABLE FUNCTION			10	10	V7R3M0	-- Description: Read t complémentaires<<
IFS	QSYS2	IFS_WRITE	PROCEDURE			10	10	V7R3M0	-- Description: Create complémentaires<<
JAVA	QSYS2	JVM_INFO	VIEW	*FILE	JVM_INFO	25	0	V7R1M0	-- Description: Find th

Services IBM i – Où trouver des exemples ?

4. En utilisant le service **SERVICES_INFO**

```
9 -- Liste de tous les services
10 SELECT COUNT(*) FROM services_info; -- 326 au 07/11/2023
11 SELECT * FROM services_info ORDER BY service_category, service_name;
```

SERVICE_CATEGORY	SERVICE_SCHEMA_NAME	SERVICE_NAME	SQL_OBJECT_TYPE	EARLIEST_POSSIBLE_RELEASE	LATEST_DB2_GROUP_LEVEL
DATABASE-UTILITY	QSYS2	SYSFILES	VIEW	V7R3M0	25
DATABASE-UTILITY	SYSTOOLS	VALIDATE_DATA	TABLE FUNCTION	V7R3M0	7
IFS	QSYS2	IFS_JOB_INFO	TABLE FUNCTION	V7R3M0	4
IFS	QSYS2	IFS_OBJECT_LOCK_INFO	TABLE FUNCTION	V7R3M0	4
IFS	QSYS2	IFS_OBJECT_PRIVILEGES	TABLE FUNCTION	V7R3M0	7
IFS	QSYS2	IFS_OBJECT_REFERENCES_INFO	TABLE FUNCTION	V7R3M0	4
IFS	QSYS2	IFS_OBJECT_STATISTICS	TABLE FUNCTION	V7R3M0	13
IFS	QSYS2	IFS_READ	TABLE FUNCTION	V7R3M0	10
IFS	QSYS2	IFS_WRITE	PROCEDURE	V7R3M0	10
IFS	QSYS2	SERVER_SHARE_INFO	VIEW	V7R3M0	10
JAVA	QSYS2	JVM_INFO	VIEW	V7R1M0	25
JAVA	QSYS2	JVM_INFO	TABLE FUNCTION	V7R4M0	25
JAVA	QSYS2	SET_JVM	PROCEDURE	V7R1M0	

Services IBM i – Où trouver des exemples ?

4. En utilisant le service **SERVICES_INFO**

```
9 -- Liste de tous les services
10 SELECT COUNT(*) FROM services_info; -- 326 au 07/11/2023
11 SELECT * FROM services_info ORDER BY service_category, service_name;
```

LATEST_DB2_GROUP_LEVEL	INITIAL_DB2_GROUP_LEVEL	EXAMPLE
25	15	-- Description: Check the reused deleted records setting for all files in APPLIB1. SELECT FILE_NAME, REUSEDLT FROM Q...
7	7	-- Description: Validate the content of FILE1. SELECT * FROM TABLE(SYSTOOLS.VALIDATE_DATA('APPLIB', 'FILE1'));
4	4	-- Description: List all the IFS references for the current job. SELECT * FROM TABLE(QSYS2.IFS_JOB_INFO(JOB_NAME ...
4	4	-- Description: List all the jobs that have a lock on /usr/test1 SELECT * FROM TABLE(QSYS2.IFS_OBJECT_LOCK_INFO(P...
7	7	-- Description: List all the authorities for objects in /usr WITH OBJS AS (SELECT PATH_NAME FROM TABLE (QSYS2....
4	4	-- Description: Look at the current reference information for /usr/test1 SELECT * FROM TABLE(QSYS2.IFS_OBJECT_REFE...
13	4	-- Description: List basic information for all the objects in directory /usr. SELECT PATH_NAME, OBJECT_TYPE, DATA_SIZ...
10	10	-- Description: Details...
10	10	-- Description: Colonnes...
10	10	-- Description: En-têtes de colonne
25	--	-- Description: Régénération
25	25	-- Description: Sauvegarde des résultats...
-	--	-- Description: Téléchargement des résultats...
20	20	-- Description: Copie
23	23	-- Description: Ctrl+C

Coller

```
-- Description: List basic information for all the objects in directory /usr.
SELECT PATH_NAME, OBJECT_TYPE, DATA_SIZE, OBJECT_OWNER
FROM TABLE (QSYS2.IFS_OBJECT_STATISTICS(START_PATH_NAME => '/usr',
SUBTREE_DIRECTORIES => 'NO'));
```

Services IBM i / DB2 – Où trouver des exemples ?

- 5. Sur le site Web IBM i Technology Update
 - <https://www.ibm.com/support/pages/node/1119129/>

Enhancement Landing Pages	
IBM i 7.5	TR3 - TR2 - TR1 - Base Enhancements
IBM i 7.4	TR9 - TR8 - TR7 - TR6 - TR5 - TR4 - TR3 - TR2 - TR1 - Base Enhancements
IBM i 7.3	TR13 - TR12 - TR11 - TR10 - TR9 - TR8 - TR7 - TR6 - TR5 - TR4 - TR3 - TR2 - TR1 - Base Enhancements
IBM i 7.2	TR9 - TR8 - TR7 - TR6 - TR5 - TR4 - TR3 - TR2 - TR1 - Base Enhancements
IBM i Memo to Users (MTU) updates	
IBM i - Technical Articles	
Db2 for i SQL Tutor	

Services IBM i / DB2 – Où trouver des exemples

- 5. Sur le site Web IBM i Technology Update

IBM i Tutorials, Demos, and SQL examples

Categories
Access Client Solutions (ACS)
Database Engineering Topics
Db2 for i Services
IBM i Services
IBM i Topics
Integrated File System (IFS)
Limits
SQL Procedural Language (SQL PL)
SQL Examples
Security

Services IBM i / DB2 – Où trouver des exemples

■ 5. Sur le site Web IBM i Technology Update

You are in: [IBM i Tutorials, Demos, and SQL examples](#) > [Integrated File System \(IFS\)](#)

GitHub Gists	Topic
IFS breakdown objects by subdir.sql Someone asked if you could count the stream file objects, by directory. This solution gets the job done and includes the total size count as well.	Number of objects in a directory
Spreadsheets and Emails with SQL & ACS.sql In this Gist, I show how you can leverage ACS's jar that is shipped on your IBM i to do your bidding. In this case, we can automate the creation and downloading of a spreadsheet to an IFS file where the contents of the spreadsheet is controlled with an SQL query. After the spreadsheet is created in the IFS, we use SQL once more to email the spre...	Save spreadsheet
Bringing order to the IFS.sql This example picks on the IFS stream files found within and under the /tmp directory. How much gunk has been accumulated under /tmp and what can you do to manage it? A bit of SQL to the rescue. The IFS_OBJECT_STATISTICS() UDTF returns many elements of data for you to leverage for improved management of the IFS.	Objects in /tmp directory
TopN user storage report.sql This example takes a previous example and extends it. The idea here is that you want to proactively manage user consumption of storage. For the top storage consumers, return a report that lists their largest objects (either in QSYS or IFS) and provide some contextual detail. This solution includes simple controls to allow the caller to specify t...	Storage
Object ownership by user - total report.sql This gist combines several IBM i (SQL) Services to produce an easy to consume UDTF. Pass in a user name and you'll get back all the QSYS and IFS objects they own, ordered by size descending.	Object ownership

Si vous avez besoin de formation

- **Workshops IBM de 4 heures**
 - SQL – Nouveautés 7.1 à 7.5
 - SQL avancé – DDL (Data Definition Language)
 - SQL avancé – DML (Data Manipulation Language)
 - SQL – XML et JSON
 - SQL – Consommation de Services Web REST
 - Création de Services Web sous IBM i avec IWS (3 workshops)
 - RPG Full Free et nouveautés 7.1 à 7.5
 - RDi (2 workshops)

- Peuvent être financés avec vos **vouchers**, sinon facturables
 - Pour connaître votre éligibilité aux vouchers, merci de me contacter (pbourgeois@fr.ibm.com)

